International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

Extended essay cover

Candidates must complete this page and then give this cover and their final version of the extended essay to their supervisor.			
Candidate session number			
Candidate name			
School name			
Examination session (May or November)	May	Year	2015

Diploma Programme subject in which this extended essay is registered: Mathematics

(For an extended essay in the area of languages, state the language and whether it is group 1 or group 2.)

Title of the extended essay: A Comparison of RSA Encryption and Diffie-Hellman Keys Exchange

Candidate's declaration

This declaration must be signed by the candidate; otherwise a mark of zero will be issued.

The extended essay I am submitting is my own work (apart from guidance allowed by the International Baccalaureate).

I have acknowledged each use of the words, graphics or ideas of another person, whether written, oral or visual.

I am aware that the word limit for all extended essays is 4000 words and that examiners are not required to read beyond this limit.

This is the final version of my extended essay.

Candidate's signature: _____ Date: 12/8/2014

Supervisor's report and declaration

The supervisor must complete this report, sign the declaration and then give the final version of the extended essay, with this cover attached, to the Diploma Programme coordinator.

Name of supervisor (CAPITAL letters) _____

Please comment, as appropriate, on the candidate's performance, the context in which the candidate undertook the research for the extended essay, any difficulties encountered and how these were overcome (see page 13 of the extended essay guide). The concluding interview (viva voce) may provide useful information. These comments can help the examiner award a level for criterion K (holistic judgment). Do not comment on any adverse personal circumstances that may have affected the candidate. If the amount of time spent with the candidate was zero, you must explain this, in particular how it was then possible to authenticate the essay as the candidate's own work. You may attach an additional sheet if there is insufficient space here.

was excited to write about aspects of cryptography, from a mathematical perspective The math involved in modern cryptography is complex and subtle.

gains focus by comparing the most-used public key cryptographic system (RSA) with a key echange technique known as Diffie-Helman. Since RSA encryption is slow, virtually all implementations use RSA encryption only to encrypt a randomly selected key, and then encrypt the bulk of the message using a (symmetric) modern cipher such as the AES standard cipher. This allows for a more-or-less straight comparison of RSA as a method to transfer a secret cipher key, on the one hand, with Diffie-Helman Key Exchange, on the other hand.

did all the research, learned all the math, got himself immersed in the vocabulary and lingo of the field; my input was largely limited to suggesting some source material up front. I later asked for various clarifications and streamlining of the material in his written drafts.

I tested 's understanding of the math involved (arithmetic modulo a very big number involving a product of large primes, for RSA – and discrete logarithms for Diffie-Helman) and I am satisfied that he understands it to a sufficient level.

Supervisor

This declaration must be signed by the supervisor; otherwise a mark of zero will be issued.

I have read the final version of the extended essay that will be submitted to the examiner.

To the best of my knowledge, the extended essay is the authentic work of the candidate.

As per the section entitled "Responsibilities of the Supervisor" in the EE guide, the recommended number of hours spent with candidates is between 3 and 5 hours. Schools will be contacted when the number of hours is left blank, or where 0 hours are stated and there lacks an explanation. Schools will also be contacted in the event that number of hours spent is significantly excessive compared to the recommendation.

I spent [2] hours with the candidate discussing the progress of the extended essay.

Supervisor's signature Date: Jan 19, 2015

Assessment form (for examiner use only)

Candidate session number		

Criteria	Achievement level				
	Examiner 1	maximum	Examiner 2	maximum	Examiner 3
A research question	1	2		2	
B introduction	0	2		2	
C investigation	2	4		4	
D knowledge and understanding	2	4		4	
E reasoned argument	2	4		4	
F analysis and evaluation	3	4		4	
G use of subject language	2	4		4	
H conclusion	0	2		2	
I formal presentation	2	4		4	
J abstract	2	2		2	
K holistic judgment	2	4		4	
Total out of 36	18				

Name of examiner 1: ___ Examiner number: __
(CAPITAL letters)

Name of examiner 2: _____ Examiner number: _____
(CAPITAL letters)

Name of examiner 3: _____ Examiner number: _____
(CAPITAL letters)

IB Assessment Centre use only: **B:** _____

IB Assessment Centre use only: **A:** _____

A Comparison of RSA Encryption and Diffie-Hellman Keys Exchange

Too broad

-Mathematics-

International Baccalaureate Program
Extended Essay

Word count: 3808

Abstract

RSA and Diffie-Hellman Key Exchange are two widely used encryption systems in modern technology. Although grounded in similar mathematics, they rely on fundamentally different problems to remain secure. The goal of the investigation was to find out which of these two methods of encryption is better under different circumstances.

I researched the capabilities, limitations, specialties, of each of DHKE and RSA, as well as how each one works. I examined different journal articles, websites, and textbooks in order to gather information on each of these encryption methods. I then used this information on the two methods to compare the two, and find out which is stronger under what conditions. After analysis, I concluded that RSA is more suitable for anything where a new key doesn't need to be created as often, such as e-mail, while DHKE is better for applications where many new keys may need to be formed for communication between many parties, such as in online communications and web browsing.

Contents

Conclusion?

You need an intro to the EE first - what is your RQ, what do you intend doing & why?

Introduction to Cryptography

Cryptography has been important in communications since the times of the Romans and the Greeks, primarily during warfare. Nowadays, cryptography plays an important role in both war and everyday life. Technology has progressed to a state where it is now to communicate and share information. This makes it much easier for information fall into the wrong hands such as those of shady corporations, real life enemies, and most worrisomely, hackers and criminals. Fortunately, along with this new technology, we have developed new cryptographic methods of keeping this information safe.

Introduction to RSA and Diffie-Hellman

RSA and Diffie-Hellman Key Exchange are two examples of modern encryption methods. Both RSA and Diffie-Hellman are unique encryption methods that that allow for secure keys to be used quickly and efficiently. The purpose of this essay is to familiarize the reader with each of these methods of encryption and their strengths, weaknesses, and specialties and then to directly compare them on each of these fronts.

This should have come first.

Too brief.

RSA

Named after Ron Rivest, Adi Shamir, and Leonard Adleman who first introduced it in 1978, RSA is a modern cipher that has stood up against years of cryptanalysis. RSA is a public key cipher, meaning that there are two keys each time it is used: one key known as the public key that anyone can know and must use in order to send a message to the receiver, who is the only

one who knows the other key, the private key. In order to generate the public and private keys, find two large prime numbers, let's call them p and q. Then find the product of these two numbers n.

$$n = p * q$$

Next, select an encryption key e such that e is relatively prime to $(p - 1) * (q - 1)$. Finally, use the Extended Euclidean algorithm to compute the decryption key d, the private key, such that

$$e * d = 1 \pmod{(p - 1) * (q - 1)} \qquad \textit{(fig 1.)}$$

Another way to state this would be

$$d = e - 1 \pmod{(p - 1) * (q - 1)}$$

The value of d can be computed using the Extended Euclidean Algorithm.

Now, you should have the values e, d, and n. Publish e and n, as people will need to use these to send you messages, but keep d a secret. This is the secret key that is needed to decipher the messages, so only you should know it. To encrypt, the sender must turn their message into a series of integers that are each less than n. The encrypted message will be represented as

similarly sized integers. The encryption formula is as follows, where m is the block of message and c is the block of encrypted message:

$$c = m^e \pmod{n}$$

To decrypt, simply use the following:

$$m = c^d \pmod{n}$$

Because:

$$c^d = (m^e)^d = m^{ed}$$

$$m^{ed} = m^{k(p-1)(q-1)+1} \quad \text{(see } \textit{fig 1.}\text{)}$$

$$m^{k(p-1)(q-1)+1} = m^{k(p-1)(q-1)} * m$$

Where is fig 1? Found it. It's equation 1 page 4

You haven't explained Modular Arithmetic sufficiently to justify this?

Prove it!

Euler's Theorem states that $a^{\varphi(n)} \equiv 1 \pmod{n}$. $\varphi(n)$ is known as Euler's Toitent Function, which counts the positive integers less than n which are also relatively prime to n. The only important thing that we need to know this for, however, is that if n is semiprime, then $\varphi(n)$ is equal to the product of one less than each of its factors. In other words

meaning

$$\varphi(n) = (p-1)(q-1)$$

Prove it!

meaning?

where $n = pq$

This is the case for our specific scenario, so we can further simplify

Too much stated without proof or justification.

$$(m^{k(p-1)(q-1)} * m) = (m^{k\varphi(n)} * m) = (1^k * m) = m$$

Because of this, the message could have been encrypted with d and decrypted with e just as easily as vice-versa.

RSA gains its security from the inherently difficult problem of obtaining the prime factorization of very large numbers which only have two prime factors. Although there is no proof, mathematicians believe that this is an inherently difficult task for which there is no possible shortcut. Similar to the conjecture that there is no shortcut to factoring large numbers, RSA has never been mathematically proven to be secure. Although many years' worth of cryptanalysis has not been able to prove or disprove the security of RSA mathematically, the lack of a solution over these many years of cryptanalysis has proved it empirically (Shneier 281).

More about RSA

RSA Speedup Using the Chinese Remainder Theorem

Many Crypto libraries use a version of a formula based upon Fermat's Little Theorem and the Chinese Remainder Theorem (CRT) in order to greatly increase the speed of decryption. The following values can be computed before receiving the encrypted message and are used in encryption:

$x = d \bmod (p - 1)$

$y = d \bmod (q - 1)$

$z = q^{-1} \bmod p$ (using the extended euclidean algorithm)

Then after receiving the encrypted message c, you can carry out the rest of the formula:

$$m_1 = c^x \bmod p$$

$$m_2 = c^y \bmod q$$

$$h = z(m_1 - m_2 + p) \bmod p$$

$$m = m_2 + hq$$

with m being your final deciphered message. This method works to speed up the process of decryption using the following equations:

$$m_1 = (c^d \bmod n) \bmod p = (c \bmod p)^{d \bmod (p-1)} \bmod p$$

The same is true replacing m_1 and p with m_2 and q respectively. This equation can be reached from our original definition of m_1 using Fermat's Little Theorem. The step between these two expressions is also what saves most of the computational time when using this algorithm. Computing the right side of this equation is much easier than computing the middle because $d \bmod (p-1)$ is so much smaller than d, fewer steps have to be taken in the process of modular exponentiation. It is also worth noting that $c \bmod n$ is much less than c, which helps speed up the process also (Paar, Pelzl 184).

Signatures in RSA

RSA can also be used to create digital signatures. To do this,encrypts m with their own private d. The receiver can decrypt this with the sender's public e. If the signature, when decrpyted, is the same as the message, then the signature is authentic (Delfs, Knebl 45).

Vulnerabilities of RSA

If two separate people have different large semiprime *n*s that share one factor, it is trivial to find the prime factorization of both *n*s by simply finding the greatest common factor of the two. It is possible to compare one large semiprime to many at the same time this way by comparing it to the product of all of those large semiprimes. Another attack, called a timing attack, finds out the message after many decipherings by how long it takes the computer to decipher each ciphertext (Paar, Pelzl 195). This can be easily thwarted by computing $(r^e c)^d$ mod n instead of just c^d mod n. This will leave you with rm mod n, from which you can find m by multiplying by the inverse of r mod n. Because RSA is also a deterministic cipher, i.e. it has no random component, a chosen plaintext attack is also possible. This method works by guessing possible plaintexts, and then enciphering them, and then comparing them to the ciphertext. Vulnerabilities to attacks such as chosen plaintext attacks and adaptive chosen ciphertext attacks can be protected against using padding (Delfs, Knebl 47). This is an extra step to encryption compared to pure RSA that adds uniformity and randomness to the cipher. It incorporates random numbers and hash functions to help do this. One strong padding method is Optimal Asymmetric Encryption Padding (RSA Laboratories). Pure RSA is very slow. Practically, users only use RSA to send the key to a more secure systematic cipher, such as Advanced Encryption Standard (AES). This allows for a much faster encryption and decryption process (Paar, Pelzl 174).

Diffie-Hellman Key Exchange

In 1976 the Diffie-Hellman Key Exchange system was published as the first public key encryption system. Unlike RSA, DHKE can only be used to securely create keys to symmetric ciphers and not to send actual messages. The usefulness of DHKE is that each participant ends up with a secret key which they both know, but nobody else knows. A good analogy has to do with different colors of paint. For this analogy pretend that it is very difficult to separate the colors of paint. Alice and Bob want to develop a secret color of paint that only they know, but any paint that sent can be intercepted by Eve, who can discover the color of the paint. Alice and Bob each start out with one quart of the same color paint, say yellow for instance. Then Alice and Bob each add of secret color paint to the yellow paint so that they each have a 2 quart mixture. The color of this secret paint doesn't matter, only that each party remembers what color it is until the end of the process, and keep the color a secret. Alice and Bob each send their 2 quart mixture to each other. Then, they add one more quart of their secret color paint to the mixture they have received, so that they each then have a 3 quart mixture. Eve knows that they each started with yellow paint because that was public, but she is unable to separate the secret paint from the yellow paint. After this, Bob and Alice each have a secret color paint that consists of one quart yellow paint, one quart of Bob's secret color paint, and one quart of Alice's secret color paint, thus they both have 3 quarts of a new secret color that they, and only they, know. However, having never known the secret color of the other party, neither Alice nor Bob could have predicted what the resulting secret color would be. DHKE works similarly to this example, but instead of a public color, there is a public modulus n and a public base e, and instead of adding a secret color paint Bob and Alice each multiply by a secret integer mod n (Vinck).

How much of this is quoted?
I'd like to see the candidate explaining/illustrating it

Imagine that you are Bob. In order to create a secret key that only you and Alice know, you first must select a large prime n and an integer e so that $1 < e < n$ and e is a *primitive root* mod n. For e to be a primitive root mod n means that for every integer a that is relatively prime to n there exists a k so that $e^k \equiv a \bmod n$. In this case k is known as the discrete logarithm for a to the base e mod n. After you, Bob, have sent these public numbers to Alice, you and she must randomly choose an integer b and a respectively, and multiply them by e mod n to get the following

What does this mean? Give an example — Show some understanding

$B = e^b \bmod n$ and

$A = e^a \bmod n$

After this you and Alice each send each other B and A. Then multiply again to get the following

$k = A^b \bmod n$ and

$k' = B^a \bmod n$

k and k' are equal, because both are equal to $e^{ab} \bmod n$. Alice and Bob can now use k, some portion of k, or even a cryptographic hash of k as the key to a symmetric cipher, most likely AES. However, k could not have been known by either party beforehand (Paar, Pelzl 206).

Not explained at all.

The reason that Diffie-Hellman is secure is because of the inherent difficulty of finding discrete logarithms modulo n. In other words it is very difficult to find b given e and n and B and $B = e^b \bmod n$ given that n is sufficiently large. It is recommended that that n be around 2048 bits. Note that there is no size limit to a and b; however they need to be at least as big as $\log_e n$. It is also recommended that the smallest prime factor of $n - 1$ be at least 256 bits in order to prevent a Pohlig-Hellman attack on the discrete log. Similar to the prime factorization problem, the discrete logarithm problem is not mathematically proven to be difficult, but so far it remains infeasible to solve for sufficiently large values of n (Paar, Pelzl 216).

Problems with DHKE

Even though this key exchange by Diffie-Hellman is supposed to be secure, the fact that there is a key exchange at all still leaves vulnerabilities. It is difficult for an eavesdropper to find Alice and Bob's private key if they can't change the intercepted messages. However, if an eavesdropper had the ability to create and delete messages (an assumption which relies on Bob and Alice not having secure digital signatures) then it is fairly easy for Eve, our eavesdropper, to know the secret key. When Eve modifies messages this way it is known as a "man in the middle" attack. Essentially Eve can replace the B with her own B' and A with her own A'. This would allow her to cause Alice and Bob to make encryption keys that correspond to her rather than with each other. This requires a lot of subsequent interception and replacement on Eve's part however, so that Alice and Bob don't realize that they actually have 2 different secret keys. One way to get around this is to instead modify the messages so that Alice and Bob end up with a very weak or known key. For instance, Eve can create a weak key if she replaces A and B with any number of the form $k^{x(n-1)}$ where $x > 1$, because this value equals one mod n. This would cause the resulting secret key to be 1 for both Alice and Bob. This would not work if Alice and

Prove it.

Bob were real people, because they would almost surely notice this difference. Realistically, though, Alice and Bob are actually probably just Alice's and Bob's computers, which might not be programmed to recognize this form of attack (Raymond, Stiglic).

Pohlig-Hellman Attack

Other attacks try to solve the discrete logarithm problem. The discrete logarithm problem can be solved relatively easily with the Pohlig-Hellman attack providing that the prime factors of n - 1 (one less than the prime modulus used for DHKE) are all fairly small. This would allow an attacker to break DHKE. Given $e = g^x \bmod p$, and the integer e, g, and p (a prime), this formula will solve for x. First, you must break up (p - 1) into its prime factors $p_1^{k1} * p_2^{k2} * p_3^{k3} \ldots p_i^{ki}$. Next we can find x modulo each of these p. To do this we will express x mod p_n as $a_n p_n + x_n$. we can find this by using the following:

$$e^{(p-1)/pn} = g^{x(p-1)/pn} \pmod p$$

$$e^{(p-1)/pn} = g^{(p-1)an} g^{(p-1)pnxn} \pmod p$$

$$e^{(p-1)pn} = g^{(p-1)pnxn} \pmod p \quad \text{(via Euler's theorem)}$$

At this point we can just guess and check every possible value of x_1 through x_i. Once you have all x_n you can find x using this generalization of the Chinese Remainder Theorem:

"For integers a_1 through a_i and relatively prime integers q_1 through q_i there exists an x such that

$x = a_1 \bmod q_1$

$x = a_2 \bmod q_a \ldots.$

$x = a_i \bmod q_i$

all solutions x are congruent modulo the product of all q_n

the following can be used to find x mod q

$x \equiv (a_1b_1q/q_1 + a_2b_2q/q_2 + ... + a_ib_iq/q_i) \bmod q$

where b_n is defined as

$b_n\, q/q_n \equiv 1 \pmod q$"

Not explained at all

The fact that this algorithm requires one to iterate through all possible values of x_n is the reason that all factors of (p - 1) must be small. As long as all p_n are small, then iterating through all possible x_n should not take very long. If there were some large p_n, and thus many possible values of x_n, however, computation of this algorithm would be infeasible. Primes of the following form are safe from this attack:

$2 * t_1 * t_2 * t_3 * \ldots * t_n + 1$

where all t are large primes (Paar, Pelzl 222).

Probabilistic Prime Testing

For both RSA and DHKE, having good random numbers and random primes is important. The secret numbers that only one person knows in DHKE, each party's secret multiplier, should be chosen randomly and not reused in order to remain secure. RSA usually requires another layer of security in the form of padding algorithms. These algorithms use hash functions and random number generation in order to make RSA stronger against certain kinds of attacks, such as chosen plaintext attacks. Also, it is very important to find the prime numbers *p* and *q* randomly, so that they are hard to guess. In fact, random number generation is important at every step of the encryption process where one has to choose new numbers, as they need to be difficult to guess. For example, even if two parties exchange a symmetric cipher key securely

using RSA, if the key is not hard to guess, then an attacker could guess it and bypass the whole RSA part altogether. Finding random primes is usually done in a guess and check method, by picking random numbers of approximately the correct size as needed and then conducting primality tests on them.

Probabilistic primality tests are actually just compositeness tests. If the test says a number is prime, there is always a small chance, however, that it really is composite. The test is repeatable for different randomly chosen "witness" integers, and for each repetition if the number is composite, there is a chance (¾ when using the Miller-Rabin test) that it will report the number as composite. Otherwise, it will make no report. If each repetition has a ¾ chance to report a composite if it is a composite, then the probability that the algorithm comprising of k repetitions of this test for different random witness numbers makes an error (i.e. doesn't report the number as a composite when it really is) is $¼^k$. One example of a strong probabilistic primality test is the Miller-Rabin test, which is computed as follows: Given a number n, choose a random witness integer $1 < a < n - 1$. Also let s and d satisfy $2^s d = n - 1$ where d is odd. For each randomly chosen a, if n is composite, there is a ¾ chance that it will return that n is composite. It will return that n is composite if

Give an example!

$a^d \not\equiv 1 \pmod{n}$ and

$a^{(2^r)d} \not\equiv -1 \pmod{n}$ for all $0 \leq r \leq s - 1$

?

notation.

This algorithm is then repeated until it returns composite or the desired number of repetitions is reached, at which point n is considered prime. It is advisable to repeat this until the error probability is below 2^{-80}, which in this case is 40 times (Paar, Pelzl 190).

Comparison of RSA and DHKE

Overall, RSA and DHKE are very similar. Both cryptographic methods rely on very difficult modular arithmetic problems. Despite this, they do have significant differences and are each better for different tasks. In terms of raw security, RSA and DHKE are somewhat similar. Both have a recommended key size of at least 2048 bits and are both somewhat vulnerable to a variety of similar attacks such as timing attacks and attacks based on poor random number generators. Commonly, both RSA and DHKE are used to create or share keys in order to use with a stronger symmetric cipher, such as AES. However, there are many differences between the two.

The biggest weakness that DHKE has over RSA is that you actually have to exchange keys. This gives attackers the opportunity to change the keys as they are being set up.

RSA is very easy to authenticate, however it is not as easy to authenticate the exchange of Diffie-Hellman keys. Security-wise, RSA seems superior to DHKE assuming that for each you have sufficient padding, random number generation, blinding to timing attacks, primes, etc. to supplement the basic algorithm. For DHKE, the primes don't need to be as strong. One only needs to be sure that the primes are strong against several fairly common discrete logarithm algorithms such as the Pohlig-Hellman algorithm. A good random number generator is recommended for generating secret exponents so that there is little correlation, but these

exponents are sometimes even safely reusable. With RSA however, it is extremely important to find primes p and q that fit certain criteria and are not too common. If an attacker guesses just one of these prime factors, then they have complete access to all messages you send until you get a new key. With RSA, usually you keep the same key for longer than with DHKE, with which the same base, prime, and secret exponents are never used again. Essentially, RSA seems stronger to a person who has all the resources and ability to ensure that the keys which they have chosen are secure, but requires more effort in order to do so. In the case that strong keys are not available for some reason, DHKE is stronger.

RSA has the advantage of utility over DHKE. DHKE can only be used for creating secret keys, and for a few other similar things such as password authenticated key agreements. RSA is useful as a public key cipher on its own in addition to use in tandem to symmetric ciphers and in digital signatures.

Computationally, both RSA and DHKE are very similar at their core. However, RSA requires a much longer key generation time. For things like email where new keys don't need to be created all the time for many different places, RSA is generally better. However, for things like online communications (i.e. web browsing), where one might go to many different websites and need new keys to communicate with multiple different parties, DHKE is generally more practical (Wiener).

The Current Value of RSA and DHKE

RSA and Diffie Hellman both seem to be on their way out. With modern computers getting stronger and stronger, it is probable that within the next ten or so years both methods will require 4096 bit keys in order to be secure. It is also quite possible that either or both of these

ciphers could even be cracked by that time. Currently the NSA is encouraging a switch from these cryptosystems to elliptic-curve cryptography (ECC), which is more secure and faster than both RSA and DHKE, as well as having a smaller secure key size of only around 128 bits (NSA). Despite this, RSA and DHKE still have a large role in contemporary security systems.

No conclusion!

An overambitious rq. Trying to compare 2 difficult ideas would have been improved by just explaining one in detail. Too many shortcuts in explaining, illustrating & showing understanding.

Criteria B, H not followed.

Works Cited

Delfs, Hans, and Helmut Knebl. *Introduction to Cryptography: Principles and Applications.* Berlin: Springer, 2002. Print.

Han Vinck, A.J. "Introduction to Public Key Cryptography." (n.d.): n. pag. 12 May 2012. Web. 29 Oct. 2014.

NSA. "The Case for Elliptic Curve Cryptography." *The Case for Elliptic Curve Cryptography.* NSA, 15 Jan. 2009. Web. 30 Oct. 2014.

Paar, Christof, and Jan Pelzl. *Understanding Cryptography: A Textbook for Students and Practitioners.* Heidelberg: Springer, 2010. Print.

Raymond, Jean-Franc, and Anton Stiglic. "Security Issues in the Diffie-Hellman Key Agreement Protocol." *Radialpoint.* Zero-Knowledge Systems Inc., 2000. Web. 30 Oct. 2014.

RSA Laboratories. "RSAES-OAEP Encryption Scheme." *RSASecurity.* RSA, 2000. Web. 29 Oct. 2014.

Schneier, Bruce. *Applied Cryptography: Protocols, Algorithms, and Source Code in C.* New York: Wiley, 1996. Print.

Singh, Simon. *The Code Book: The Evolution of Secrecy from Mary, Queen of Scots, to Quantum Cryptography.* New York: Doubleday, 1999. Print.

Wiener, Michael J. "Performance Comparison of Public-Key Cryptosystems."*CryptoBytes* 4 (1998): 1-5. Web. 30 Oct. 2014. <ftp://ftp.rsasecurity.com/pub/cryptobytes/crypto4n1.pdf>.

Works Cited

- S. Ager, *English language, alphabet and pronunciation*, Omniglot, 2014, available at `http://www.omniglot.com/writing/english.htm`.
- C. Choffrut and K. Culik, *On extendibility of unavoidable sets*, Discrete Appl. Math. **9** (1984), 125-137.
- V. Dare and R. Siromoney, *Subword topology*, Theoret. Comput Sci. **47** (1986), 159-168.
- P. Higgins and C. Saker, *Unavoidable sets*, Theoret. Comput. Sci. **359** (2004), 231-238.
- P. Huy and N. Huyen, *Generating of Minimal Unavoidable Sets*, Vietnam J. Math. **34** (2006), 459-472.
- P. Huy and N. Lam, *Unavoidable set: extension and reduction*, RAIRO Inform. Thor. Appl. **33** (1999), 213-225.
- L. Rosaz, *Inventories of unavoidable languages and the word-extension conjecture*, Theoret. Comput. Sci. **201** (1998), 151-170.